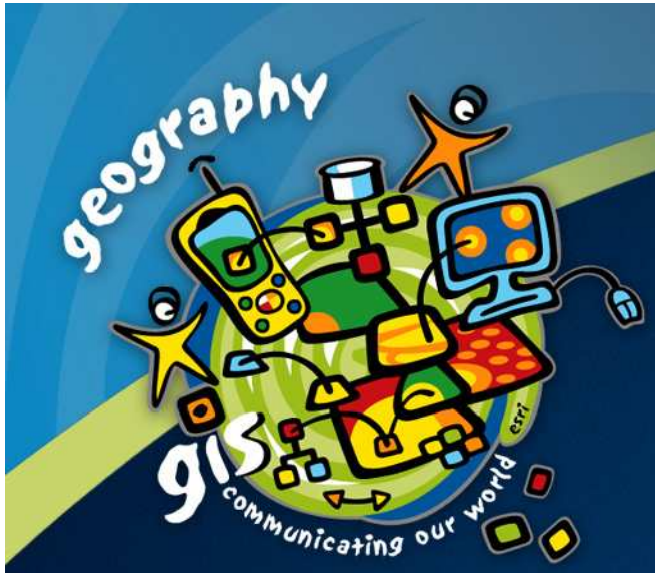


Creating custom Web Services with ArcGIS Server

UGIC 2008

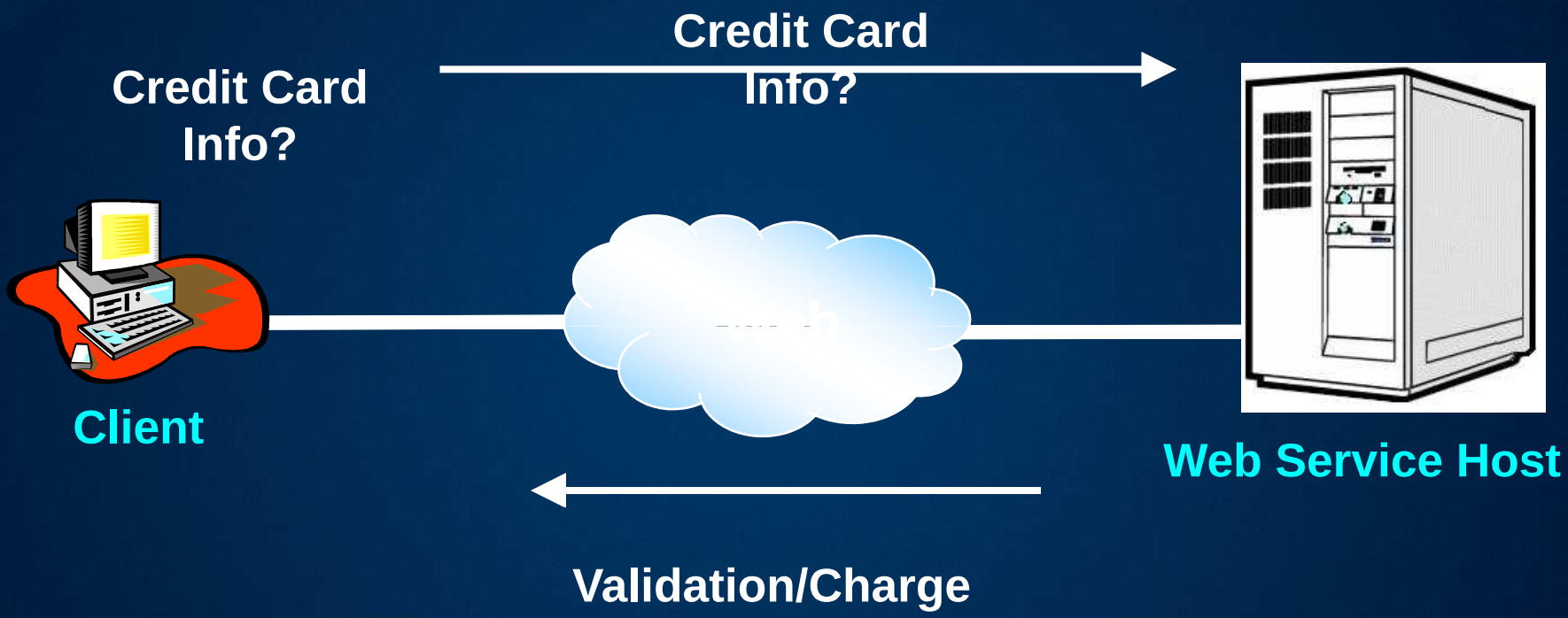
Jeremiah Lindemann

Agenda

- ArcGIS Server overview
- Services that ArcGIS Server provide
- How to create a web service
  - Tie your web service into ArcGIS Server to perform GIS Analysis
- Building a client that consumes the web service
- Question and Answer

What is a Web service?

- Functionality accessed over the Internet



A

GIS Is Evolving—and Supporting the Enterprise

Exploiting Advancing Infrastructure, Architecture, and Application Technologies …

Legacy

Client / Server

Today

Web Services

(Mapping & Visualization)

Emerging

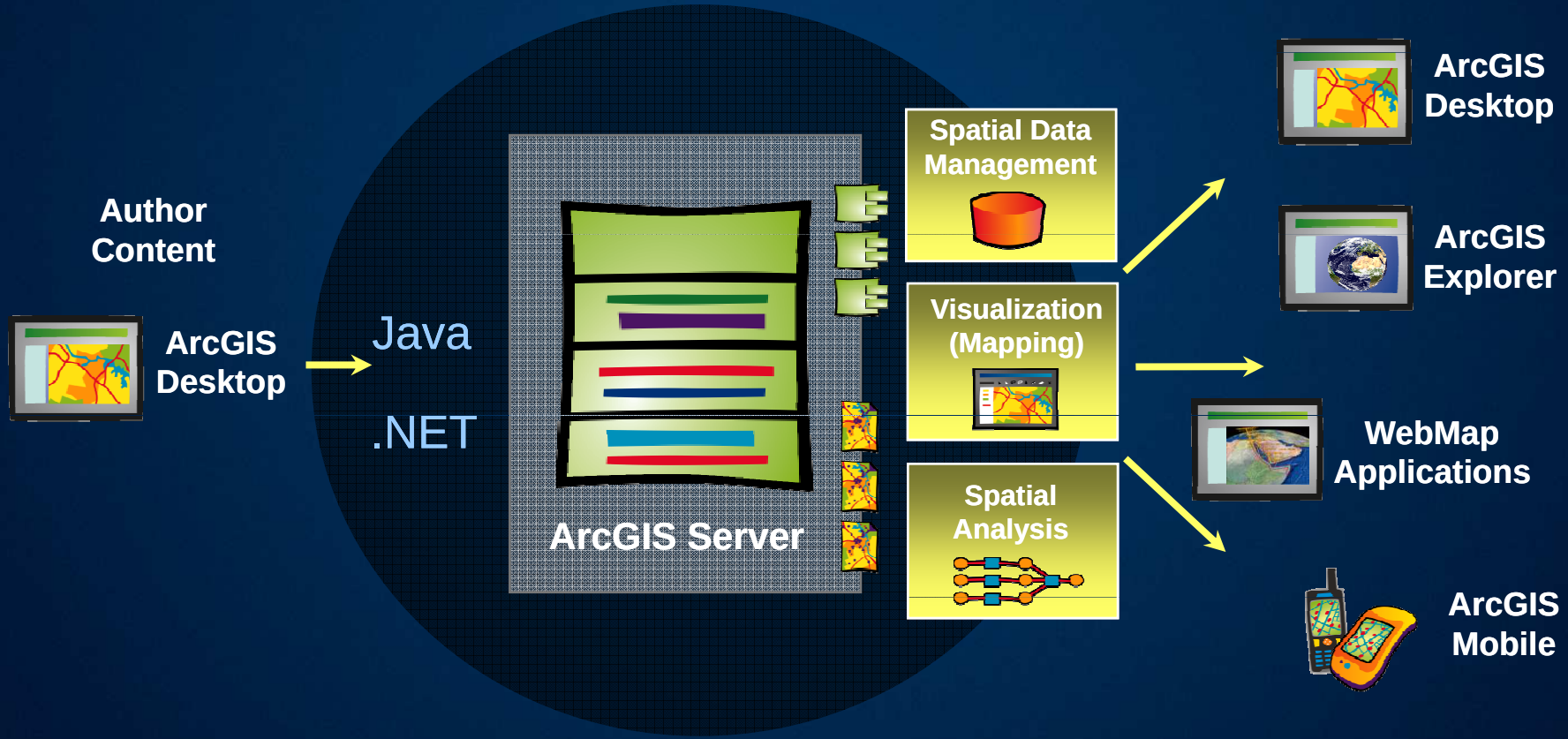
Services Oriented Architecture

- Integrating Distributed Data and Applications
- Connecting Communities and Enabling Collaboration
- Supporting Real-time Operations
- Powering the Specialist—and Reaching the Generalist

. . . Entering the Mission Mainstream—a Foundation for Analysis and Decision-making

ArcGIS Server 9.2

- Complete & Integrated server-based GIS
- Out-of-the-box applications and services
- Rich developer opportunities



GIS Services

- GIS Services are the building blocks for
  - Web applications
  - ArcGIS Explorer maps
  - Mobile applications

- Types of GIS Services that can be consumed in 9.2
  - Map Service (2D)
  - Globe Service (3D)
  - Geocoding Service
  - Network Analysis Service
  - Geoprocessing Service
  - Geodata Service
  - Mobile Data Service
  - KML
  - WMS

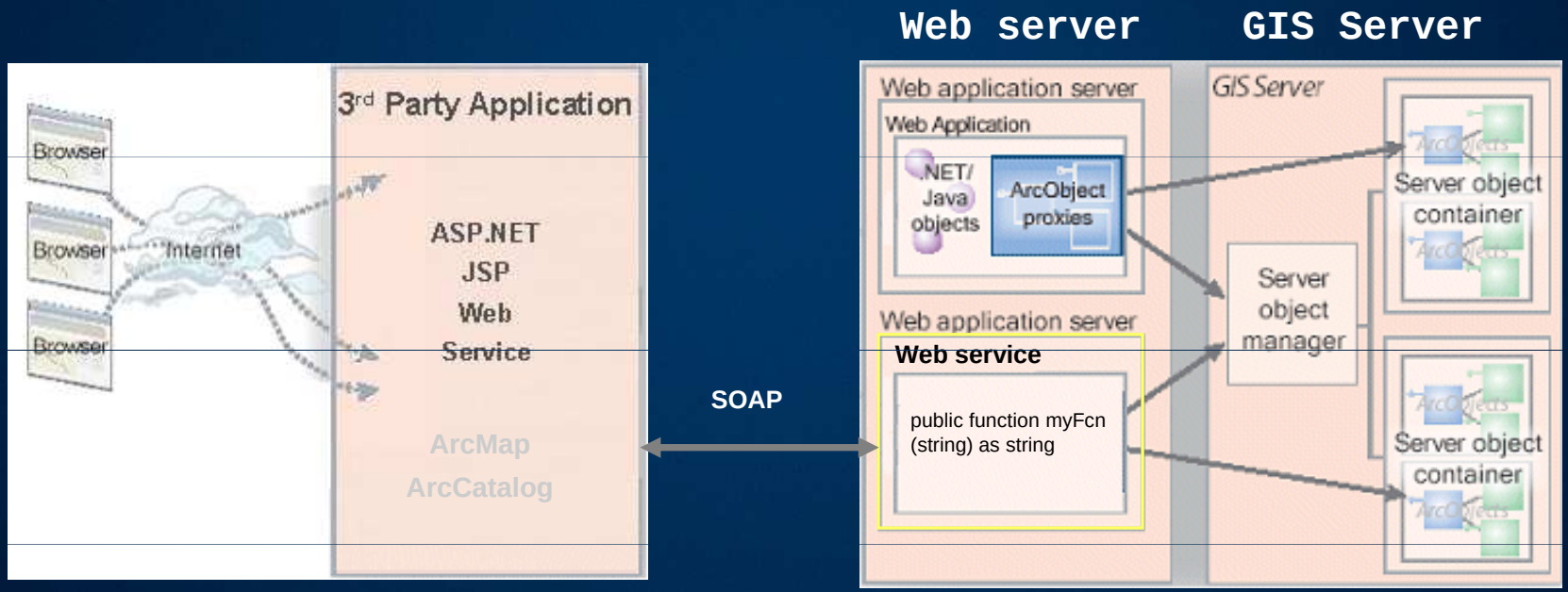
What about a web service that does something specific for your needs?

- What is the attribute value at a given X/Y location?
- What is the X/Y location (centriod) of a given Parcel?
- Return all the ATMs within a certain buffer distance of an address

- All possible via your own custom web service
  - ArcGIS Server services need not be configured
  - Create a regular web service and incorporate GIS functionality

Where do Web services fit into ArcGIS Server?

- Web Services are developed on the Web server
- Third-party applications consume the Web service
- Web service still consumes the ADF and works with the SOM



Agenda

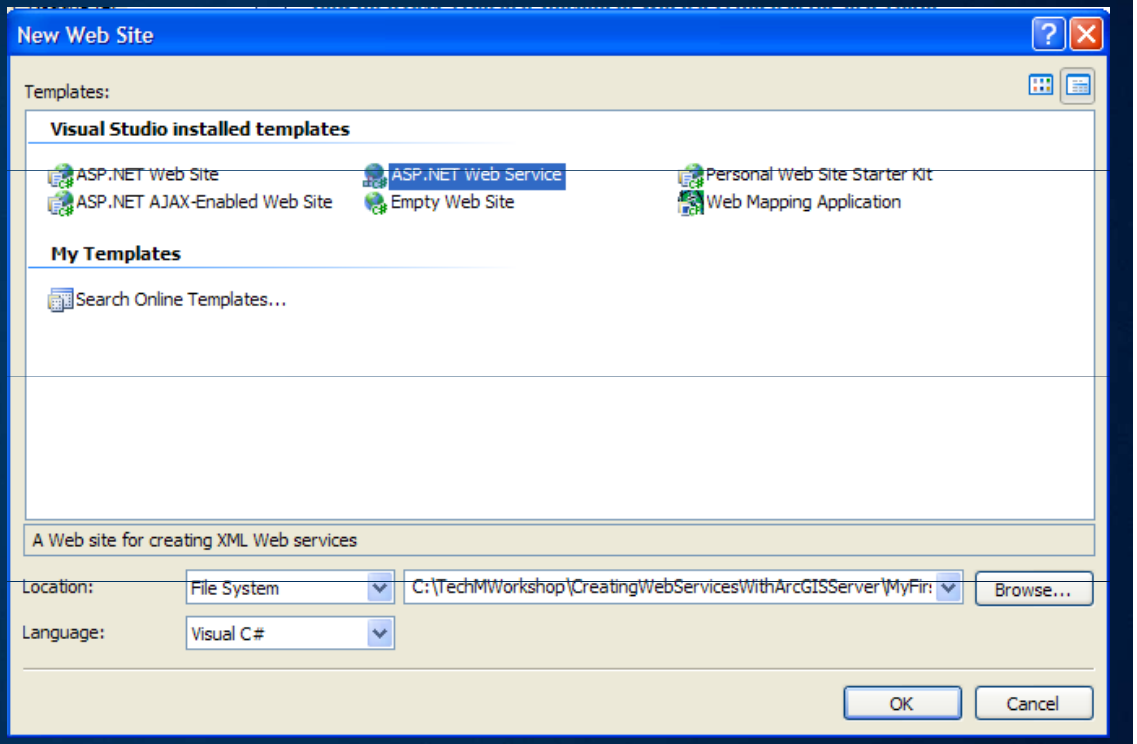
- ArcGIS Server overview
- Services that ArcGIS Server provide
- How to create a web service
  - Tie your web service into ArcGIS Server to perform GIS Analysis
- Building a client that consumes the web service
- Question and Answer

Building Web services with ArcGIS Server

- General steps
  1. Create ASP.NET Web service application
  2. Reference ESRI assemblies
  3. Create the Web service method
  4. Test the Web service

Create an ASP.NET Web service application

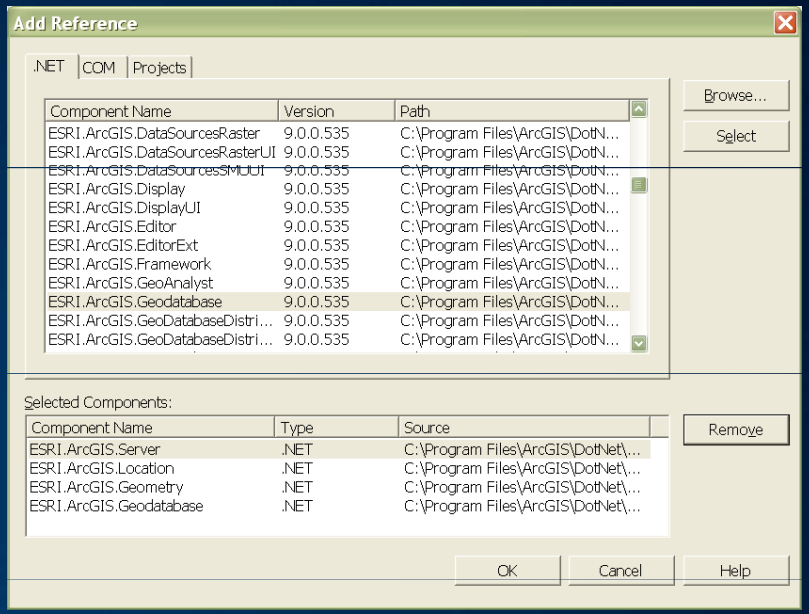
- Develop the service on the Web application server machine
- Requirements
  - ArcGIS Server ADF must be installed



Reference ESRI assemblies

- Reference necessary assemblies
- Project > Add Reference or Project > Add ArcGIS Reference
  - ESRI.ArcGIS.Server
  - ESRI.ArcGIS.ADF
  - ESRI.ArcGIS.ADF.Connection

  Other references depend on what classes you utilize:
  - ESRI.ArcGIS.xxx
- Add Imports/Using statements to class




```
Imports ESRI.ArcGIS.ADF.Connection
Imports ESRI.ArcGIS.Server
'References for Specific ArcGIS work
Imports ESRI.ArcGIS.Geometry
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.DataSourcesGDB
```

Implement impersonation

- Two ways to implement impersonation for Web services
  - Impersonate as user in the agsusers group
- Edit the Web.config file

```
<identity impersonate= "true" userName= “user" password= “pwd"/>
```

- Use the Identity class from the ArcGIS Server ADF
  - Most common methodology

```
Dim username As String = “username"
        Dim password As String = “pwd"
        Dim domain As String = “domain"
        Dim host As String = “localhost"

        'Create the user identity
        Dim userIdentity As ESRI.ArcGIS.ADF.Identity =
  New Identity(username, password, domain)
```

Implement the Web method

- Define Function/WebMethod name
- Input parameters are what user will enter
- Return value- may hold more complex data types such as an array

```
<WebMethod()> _
Public Function GetParcelID(ByVal x As Double, ByVal y As Double) As String

End Function
```

Make a connection to ArcGIS Server

- Add Identity
- Use AGSServerConnection to connect to ArcGIS Server
  - Gateway to accessing GIS functionality in Web Service

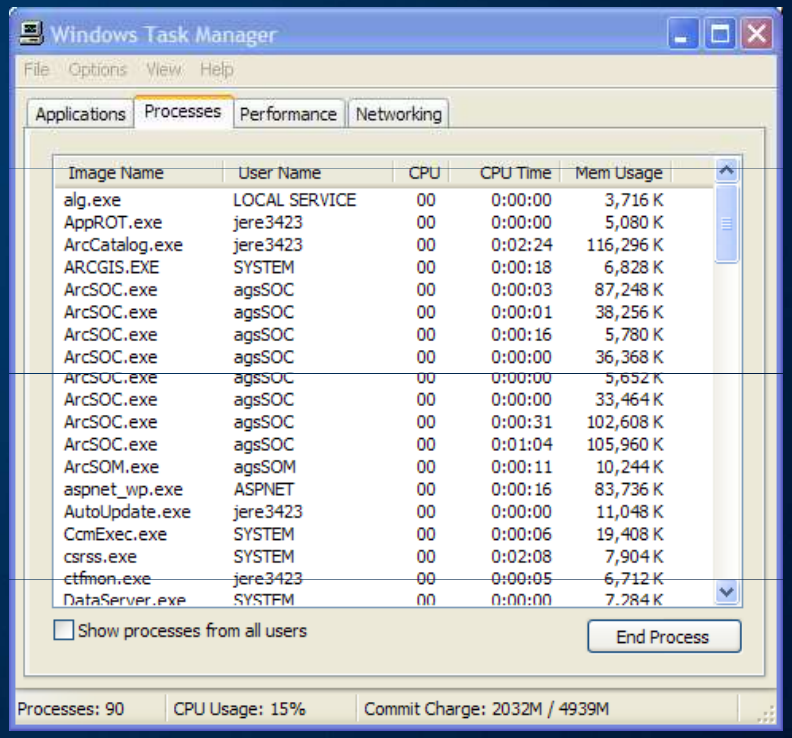
```
<WebMethod()> _
    Public Function GetParcelID(ByVal x As Double, ByVal y As Double) As String
        Dim username As String = “username"
        Dim password As String = “pwd"
        Dim domain As String = “domain"
        Dim host As String = “localhost"

        'Create the user identity.
        Dim userIdentity As ESRI.ArcGIS.ADF.Identity = New Identity(username, password, domain)
        'Create a connection object to an ArcGIS Server (host), with user credentials.
        Dim agsConnnection As AGSServerConnection = New AGSServerConnection(host, userIdentity)
        'Connect to the ArcGIS Server.
        agsConnnection.Connect()

        Return "Test"
  End Function
```

What is a server context?

- Server context is a handle to a process (ArcSOC.exe)
  - Can either create context new or obtain context from existing GIS service (map service
- Acts as a gateway to the server object and ArcObjects
- Need to manage the lifetime of server context carefully!
  - Call ReleaseContext method



Access/Create a server context

- Access SOM to create Server Context
- Server Context = processing space for ArcObjects
- Can either create empty context or use context from an existing map service

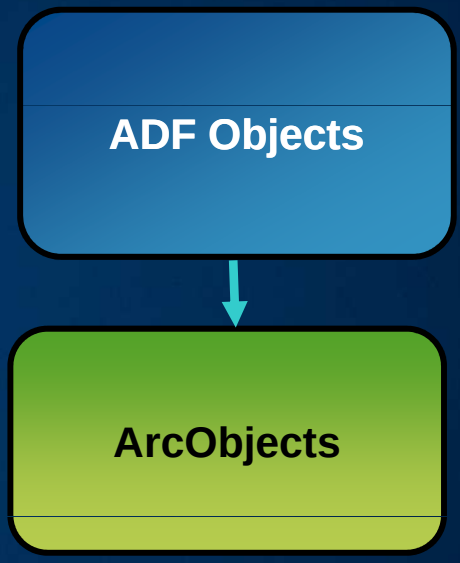
```
<WebMethod()> _
    Public Function GetParcelID(ByVal x As Double, ByVal y As Double) As String
        ……
        …… 'Continuing code from previous slide
        'Connect to the ArcGIS Server.
        agsConnnection.Connect()

        'Get the ServerObjectManager.
        Dim som As IServerObjectManager = agsConnnection.ServerObjectManager
        'A Server context is a processing space to do work
        'in the future, may want to grab the server context of an existing Map Service
        'right now, just reading data from disk, not a map service
        Dim serverContext As IServerContext
        serverContext = som.CreateServerContext("", "")
                    'ArcObjects/Analysis code will go here
        serverContext.ReleaseContext() 'Important to release context

        Return "Test"
  End Function
```

Programming GIS functionality

- Two sets of Objects: ADF and Fine Grained ArcObjects
- If fine grained GIS functionality doesn't exist at ADF level, you may need to use ArcObjects
  - May use a combination
  - Will use ADF if querying/analyzing against an existing map service
  - If creating empty context, can go straight to ArcObjects



http://edndoc.esri.com/arcobjects/9.2/NET_Server_Doc/developer/getting_started.htm

Creating Fine Grained ArcObjects in server contexts

- All ArcObjects objects need to be created in the context of the server
- IServerContext
  - CreateObject: Pass in the ProgID of the coclass
- Creates object on the server side
- Never call *New* when declaring ArcObjects types!

```
Dim pt as IPoint
Set pt = New PointClass
```

```
Dim pSCX As IServerContext
Set pSCX = pSOM.CreateServerContext("", "")
Dim pt as IPoint
Set pt = pSCX.CreateObject("esriGeometry.Point")
' Do something with the point
...
pSCX.ReleaseContext
```

Perform the GIS Analysis

- The rest is GIS coding!
- Code below uses Fine Grained ArcObjects (server context)
- May be able to use ADF object to perform analysis

```
<WebMethod()> _
    Public Function GetParcelID(ByVal x As Double, ByVal y As Double) As String
        serverContext = som.CreateServerContext("", "")
        …… 'Continuing code from previous slide
        'Create a point object to store the geometry
        Dim pnt As IPoint = serverContext.CreateObject("esriGeometry.Point")
        pnt.PutCoords(x, y)

        'Reference the Parcels FeatureClass from the gdb
        Dim wsFactory As IWorkspaceFactory =
  serverContext.CreateObject("esriDataSourcesGDB.FileGDBWorkspaceFactory")
        Dim ws As IFeatureWorkspace =
  wsFactory.OpenFromFile("C:\Data\FtCollins\FtCollinsFileGDB.gdb", 0)
        Dim fc As IFeatureClass = ws.OpenFeatureClass("Parcels")

        'Perform the Spatial Analysis by interesting the point
        Dim spatialFilter As ISpatialFilter =
  serverContext.CreateObject("esriGeoDatabase.SpatialFilter")
        …….
  End Function
```

Code sample of finding parcel ID

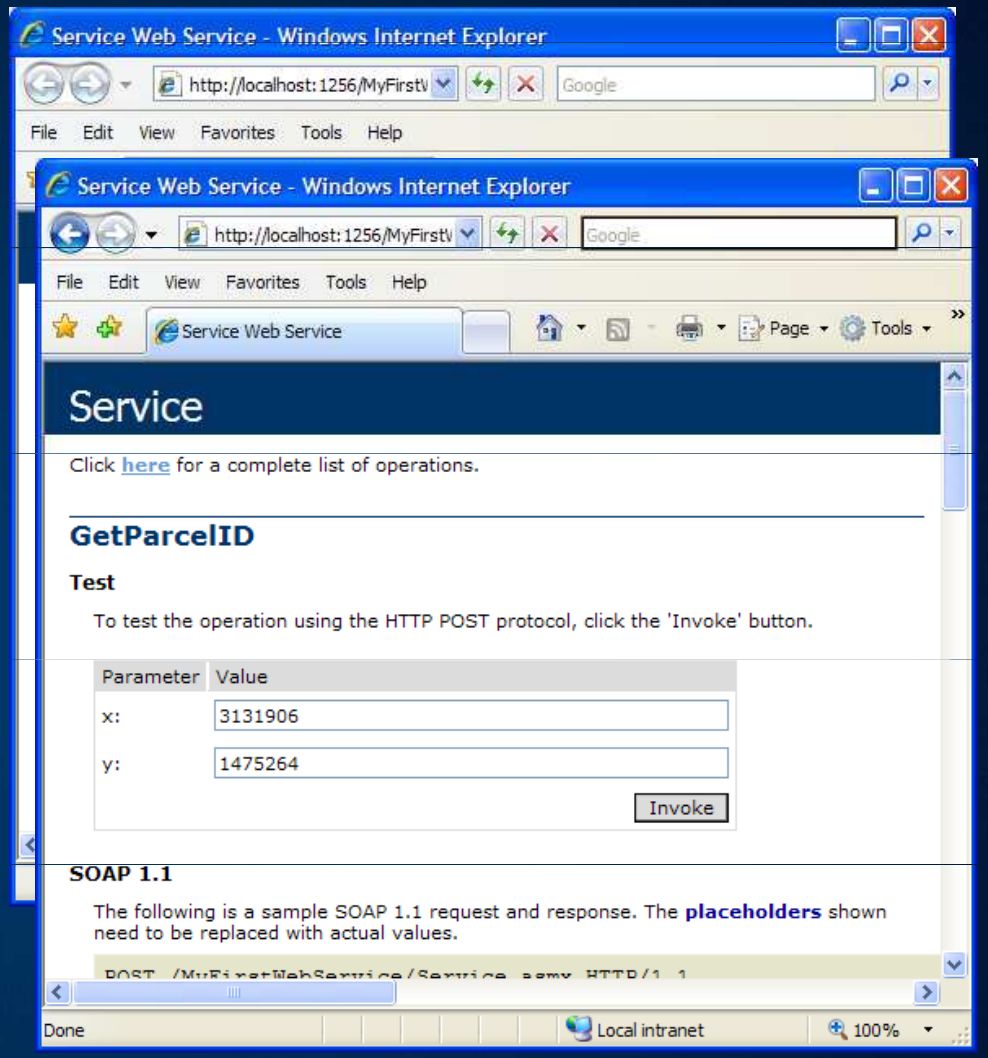
```
<WebMethod()> _
    Public Function GetParcelID(ByVal x As Double, ByVal y As Double) As String
       'Create the user identity.
        Dim userIdentity As ESRI.ArcGIS.ADF.Identity = New ESRI.ArcGIS.ADF.Identity(“username”, “pwd”, “domain”)
        'Create a connection object to an ArcGIS Server (host), with user credentials.
        Dim agsConnnection As ESRI.ArcGIS.ADF.Connection.AGS.AGSServerConnection = New AGSServerConnection(“host”, userIdentity)
        'Connect to the ArcGIS Server.
        agsConnnection.Connect()
        'Get the ServerObjectManager.
        Dim som As IServerObjectManager = agsConnnection.ServerObjectManager
        'A Server context is a processing space to do work
        Dim serverContext As IServerContext
        serverContext = som.CreateServerContext("", "")
        'Create a point object to store the geometry
        Dim pnt As IPoint = serverContext.CreateObject("esriGeometry.Point")
        pnt.PutCoords(x, y)
        'Reference the Parcels FeatureClass from the gdb
        Dim wsFactory As IWorkspaceFactory = serverContext.CreateObject("esriDataSourcesGDB.FileGDBWorkspaceFactory")
        Dim ws As IFeatureWorkspace = wsFactory.OpenFromFile("C:\Data\FtCollins\FtCollinsFileGDB.gdb", 0)
        Dim fc As IFeatureClass = ws.OpenFeatureClass("Parcels")
        'Perform the Spatial Analysis by interesting the point
        Dim spatialFilter As ISpatialFilter = serverContext.CreateObject("esriGeoDatabase.SpatialFilter")
        ' Set the Geometry to the buffer we created earlier.
        spatialFilter.Geometry = pnt
        'Set the GeometryField to the ShapeFieldName of the FeatureClass.
        spatialFilter.GeometryField = fc.ShapeFieldName
        'Specify an Intersection spatial relationship.
        spatialFilter.SpatialRel = ESRI.ArcGIS.Geodatabase.esriSpatialRelEnum.esriSpatialRelIntersects
        'Get a FeatureCursor to enumerate the features filtered by our SpatialFilter.
        Dim fCursor As IFeatureCursor = fc.Search(spatialFilter, True)
        Dim ParcelFieldIndex As Integer = fc.Fields.FindField("PARCELNO")
        Dim feature As IFeature = fCursor.NextFeature()
        If feature Is Nothing Then
            Return "No Parcel was found at that X/Y location"
        Else
            Dim strParcelNo As String = feature.Value(ParcelFieldIndex).ToString()
            Return "Parcel Number = " & strParcelNo
        End If
       'Release the Server Context. !!!! IMPORTANT !!!!!
        serverContext.ReleaseContext()
    End Function
```

Testing a Web service in Visual Studio .NET

- Steps
  1. Run Visual Studio .NET
  2. Web browser lists description and methods available
- Service Description
  - Formatted XML describing the service
- WebMethod Appears
  - Can test methods in browser
  - Illustrates SOAP requests for method

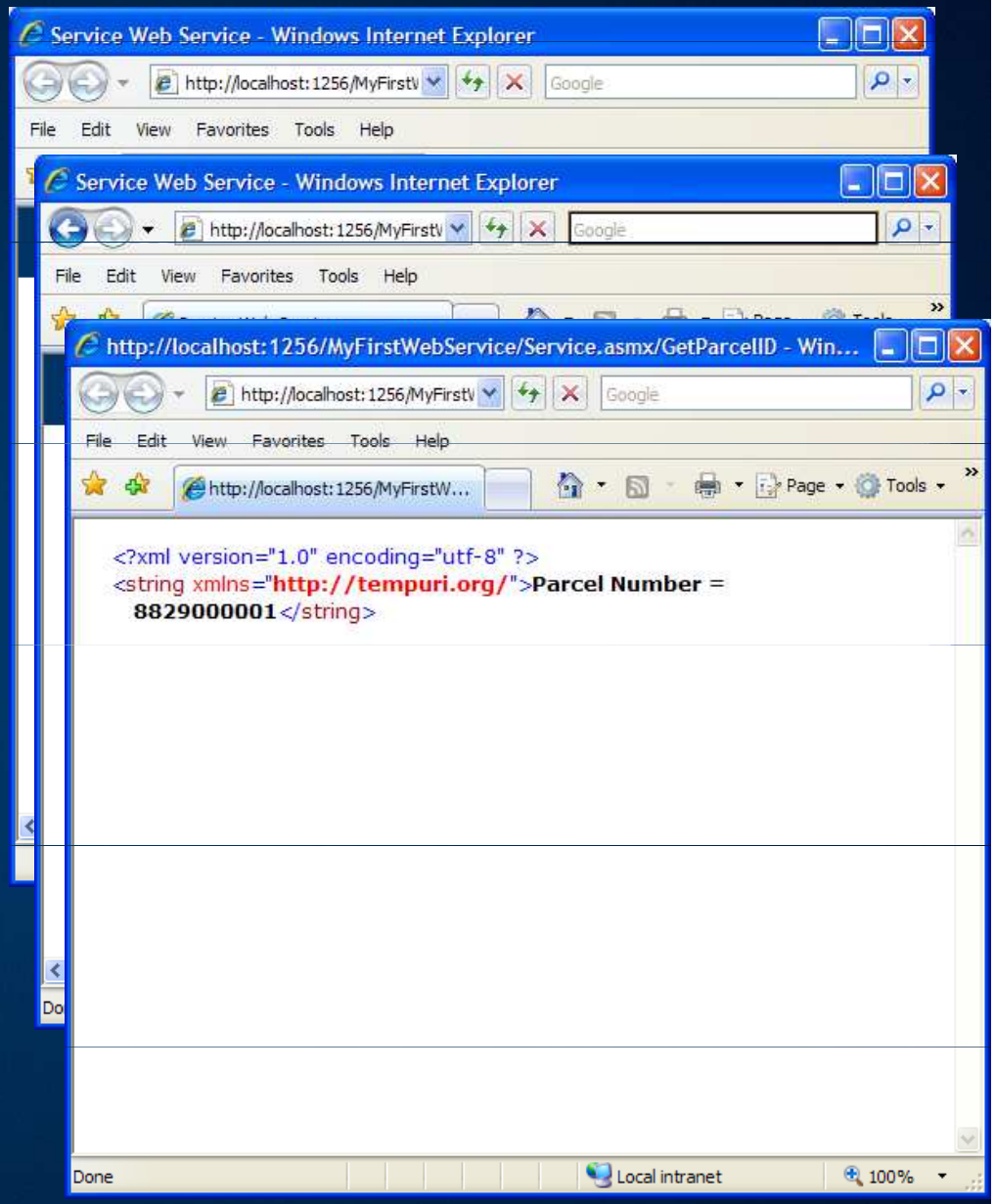
Testing a Web service in Visual Studio .NET

- Enter parameter information
- Click Invoke



Testing a Web service in Visual Studio .NET

- XML page is displayed with value returned from web service



Agenda

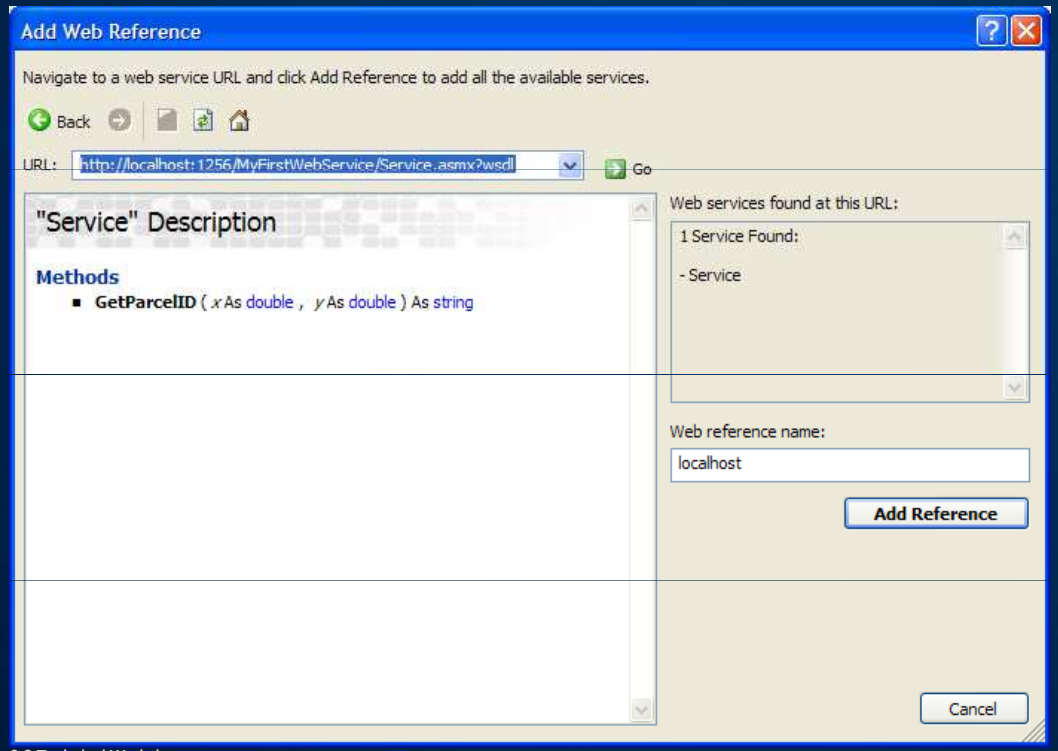
- ArcGIS Server overview
- Services that ArcGIS Server provide
- How to create a web service
  - Tie your web service into ArcGIS Server to perform GIS Analysis
- Building a client that consumes the web service
- Question and Answer

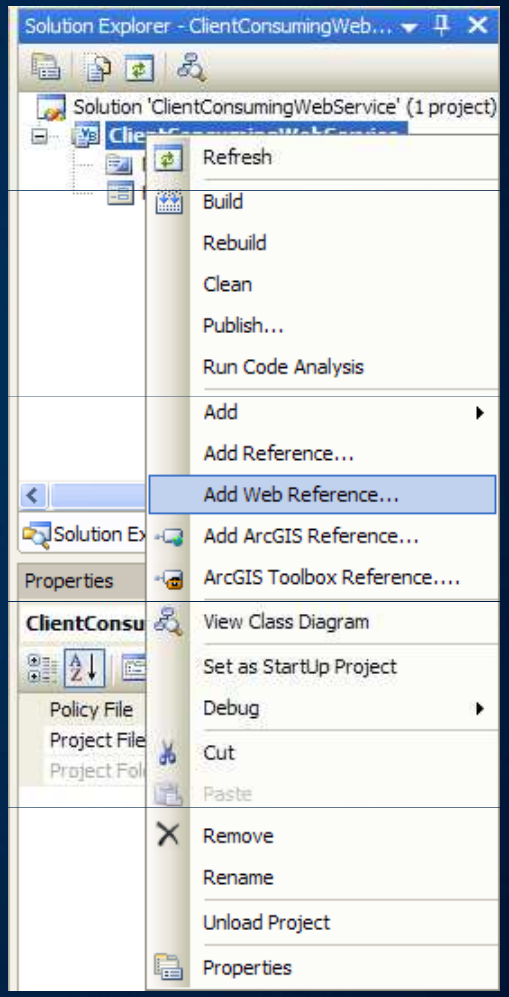
Building a client to consume Web services

- Any type of client can consume web service
  - Web page, stand alone exe, ArcGIS Explorer, ArcGIS Desktop
- All Web services publish a WSDL
- A WSDL describes the SOAP requests and responses
  - E.g., http://localhost/WebService1/Service1.asmx?wsdl
- Steps
  1. Create an application
  2. Reference a WSDL
  3. Consume the Web service functionality

Building a client to consume Web services

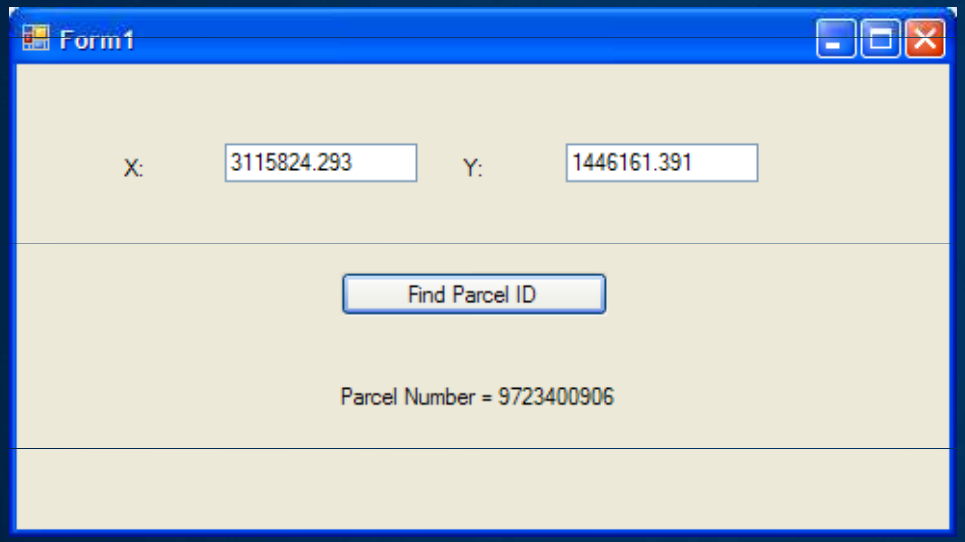
- Steps 1 & 2: Build an Application/Reference a WSDL
  - Click GO, then click Add Reference
    - May want to change Web reference name (localhost)
      - Will be how you programmatically connect

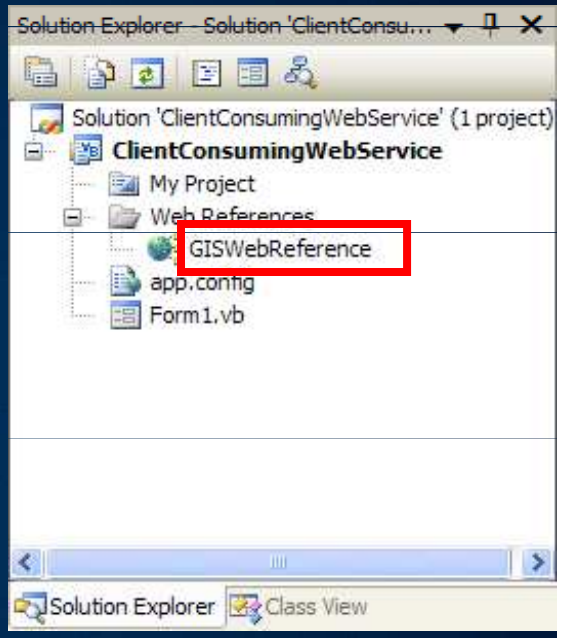




Building a client to consume Web services

- Steps 3: Consume the Web Service Functionality
  - Declare a variable to reference to the web reference
  - All methods will be exposed






```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
  Handles Button1.Click
        Dim mywebservice As GISWebReference.MyFirstWebService = New
  GISWebReference.MyFirstWebService
        lblID.Text = "Parcel ID = " & mywebservice.GetParcelID(txtX.Text, txtY.Text)
End Sub
```

Example

- Use Geocoding Service to create context
- User enters address, returns XY

Code sample of using Geocoding Server to get XY

```
<WebMethod()> _
    Public Function ReturnGeocodeXY(ByVal StreetAddress As String) As String
        'Create the user identity.
        Dim userIdentity As ESRI.ArcGIS.ADF.Identity = New ESRI.ArcGIS.ADF.Identity(username,
  password, domain)
        'Create a connection object to an ArcGIS Server (host), with user credentials.
        Dim agsConnnection As ESRI.ArcGIS.ADF.Connection.AGS.AGSServerConnection = New
  ESRI.ArcGIS.ADF.Connection.AGS.AGSServerConnection(host, userIdentity)
        'Connect to the ArcGIS Server.
        agsConnnection.Connect()

        'Get the ServerObjectManager.
        Dim som As IServerObjectManager = agsConnnection.ServerObjectManager
        'Get Server Context from existing Geocoding Service
        Dim serverContext As IServerContext
        serverContext = som.CreateServerContext("FtCollins/Composite_Address_Locator",
  "GeocodeServer")

        Dim so As IServerObject = serverContext.ServerObject
        Dim gc As IGeocodeServer = so
        Dim ps As IPropertySet = serverContext.CreateObject("esriSystem.PropertySet")
        ps.SetProperty("Street", StreetAddress)
        Dim res As IPropertySet = gc.GeocodeAddress(ps, Nothing)
        Dim gcPoint As IPoint = res.GetProperty("Shape")

        If gcPoint Is Nothing Then
            Return "Address not found"
        Else
            Return gcPoint.X & " " & gcPoint.Y
        End If
    End Function
```

Use an existing service to perform analysis

- Examples
  - Geocoding Service to return point locations
  - Map Service to query multiple layers within map service
  - Map Service to perform network analysis and return driving directions
  - Geodata services to process data
- Use existing service to obtain server context

```
<WebMethod()> _
   Public Function ReturnGeocodeXY(ByVal StreetAddress As String) As String
       '……Existing Identity code above
      'Create a connection object to an ArcGIS Server (host), with user credentials.
       Dim agsConnnection As ESRI.ArcGIS.ADF.Connection.AGS.AGSServerConnection = New
 ESRI.ArcGIS.ADF.Connection.AGS.AGSServerConnection(host, userIdentity)
       'Connect to the ArcGIS Server.
       agsConnnection.Connect()

       'Get the ServerObjectManager.
       Dim som As IServerObjectManager = agsConnnection.ServerObjectManager
       'Get Server Context from existing Geocoding Service
       Dim serverContext As IServerContext
       serverContext = som.CreateServerContext("mygeocodeservice", "GeocodeServer")
```

Questions?

- Thanks for your time!